

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH LINEÁRNÍCH STRUKTUR NA TRANZISTOROVÉ ÚROVNI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ADAM GAJDOŠ

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH LINEÁRNÍCH STRUKTUR NA TRANZISTOROVÉ ÚROVNI

DESIGN OF LINEAR STRUCTURES ON TRANSISTOR LEVEL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ADAM GAJDOŠ

VEDOUcí PRÁCE
SUPERVISOR

Ing. JAROSLAV KOTON, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Adam Gajdoš

ID: 134479

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Návrh lineárních struktur na tranzistorové úrovni

POKYNY PRO VYPRACOVÁNÍ:

V prostředí Mathematica algoritmizujte modifikovanou metodu uzlových napětí. Jako aktivní prvky využívejte tranzistory řízené elektrickým polem, které budou popsány vhodným modelem. Využití vytvořeného algoritmu ukažte na návrhu lineárních funkčních bloků s MOS tranzistory a kapacitami.

DOPORUČENÁ LITERATURA:

[1] Mathematica Tutorial: Core Language - Wolfram Mathematica® Tutorial Collection, ISBN: 978-1-57955-052-3, 2008.

[2] Mathematica Tutorial: Mathematics and Algorithms - Wolfram Mathematica® Tutorial Collection, ISBN: 978-1-57955-054-7, 2008.

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Jaroslav Koton, Ph.D.

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Bakalářská práce se zabývá vývojem software pro automatizovaný návrh kmitočtových filtrů s použitím aktivních prvků místo induktorů.

První část práce se zabývá rozdělením kmitočtových filtrů podle přenášeného spektra a podle druhu použitých součástek. Dále se zabývá taky pracovními režimy obvodu.

V další části prouzkoumává různé návrhové metody filtrů s aktivními prvky, soustředí se hlavně na metodu autonomních obvodů, která se používá programovým řešením.

Práce se dále zabývá popisem prostředí Wolfram Mathematici, jejím vývojem a taky výhodami.

V další kapitole se práce věnuje teoretickému popisu programu a všem nutným teoretickým ustanovením, které jsou potřeba ke zvládnutí programu, tj. popis Modifikované metody uzlových napětí, nahrazení tranzistoru MOSFET za zdroj proudu řízený napětím a taky zápisu pasivních prvků ve strojové podobě.

Poslední kapitola se věnuje samotnému vývoji programu a jeho podrobnému popisu.

KLÍČOVÁ SLOVA

Wolfram Mathematica, MOSFET tranzistor, kondenzátor, aplikace, autonomní obvod, filtr

ABSTRACT

Bachelor thesis deals with the development of software for automated design frequency filters using active elements instead of inductors.

The first part deals with the allocation of frequency filters by transmitted spectrum and type of components used. It also deals with working mode in the circuit.

Thesis also focuses on the description development environment in Wolfram Mathematica, and overall development and also advantages.

The next chapter is devoted to the theoretical description of the program and all the necessary theoretical provisions, that are needed to manage the program, i.e. description modified nodal analysis, replacement of MOSFET transistor for Voltage Controlled Current Source and writing passive elements in programming language too.

The last chapter is dedicated to the development of the program and detailed description.

KEYWORDS

Wolfram Mathematica, MOSFET transistor, capacitor, application, autonomous circuit, filter

GAJDOŠ, Adam *Návrh lineárních struktur na tranzistorové úrovni*: bakalářská práce. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 41 s. Vedoucí práce byl Ing. Jaroslav Koton, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh lineárních struktur na tranzistorové úrovni“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

BRNO

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Jaroslavu Kotonovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

BRNO

.....

(podpis autora)

OBSAH

Úvod	10
1 Kmitočtové filtry	11
1.1 Základní druhy kmitočtových filtrů	11
1.2 Pracovní režim obvodu	13
2 Metody návrhu filtrů s aktivními prvky	16
2.1 Návrh pomocí metody autonomních obvodů	16
2.2 Návrh pomocí úplné admitanční sítě	17
2.3 Rozšiřování autonomních obvodů	17
2.4 Syntéza filtrů pomocí grafů signálových toků	18
2.5 Návrh filtrů pomocí stavového popisu	19
2.6 Přidružená transformace	20
2.7 Využití syntetických prvků s imitancí vyšších řádů pro návrh kmitočtových filtrů	21
3 Wolfram Mathematica	24
3.1 Využití, historie	24
3.2 Prostředí Mathematici	25
4 Automatizace návrhu lineárních obvodů	26
4.1 Princip modifikované metody uzlových napětí	26
4.2 Tranzistor MOSFET a jeho nahrazení zdrojem proudu řízeného napětím (VCCS)	26
4.3 Popis automatizace návrhu autonomního obvodu	27
4.4 Teoretické řešení realizace automatizovaného návrhu	28
5 Programové řešení	30
5.1 Podprogram - NoveZadani.nb	30
5.2 Hlavní program - HlavniProgram.nb	32
5.3 Podprogram - navyseniM.nb	35
6 Závěr	37
Literatura	38
Seznam symbolů, veličin a zkratek	40
Seznam příloh	41

SEZNAM OBRÁZKŮ

1.1	ideální filtry a) DP b) HP c) PP d) PZ	12
1.2	reálné filtry a) DP b) HP c) PP d) PZ	13
1.3	obvod v napěťovém módu	14
1.4	obvod v proudovém módu	14
1.5	Obvod ve smíšeném módu	15
2.1	a) obecný autonomní obvod, b) proudový mód	17
2.2	a) původní, b) zjednodušené zobrazení prvku MO-CF	19
2.3	Kaskádní zapojení neinvertujících integračních článků se zpětnými vazbami	19
2.4	Obvod pracující v napěťovém módu	20
2.5	Ukázka aplikace přidružené transformace, obvod pracující v proudov- ovém módu	21
2.6	Syntetické dvojpóly typu a), DP, b) EP, c) DS, d) ES [12]	22
2.7	Obecná realizace syntetických prvků vyšších řádů a) sériové připojení vnější admitance b) paralelní připojení vnější admitance c) bez nut- nosti připojení vnější admitance [7]	23
3.1	prostředí Mathematici	24
3.2	ukázka práce Kernelu [13]	25
4.1	a) tranzistor MOSFET b) náhradní model tranzistoru pomocí VCCS	27
4.2	Př. zapojení pasivní sítě (<code>code-p = 12202334</code>)	28
4.3	vývojový diagram programu	29
5.1	část vstupního souboru <i>zadani.txt</i>	30

SEZNAM TABULEK

2.1	Souhrn elementárních dvojpolů	21
4.1	razítko VCCS	27

ÚVOD

Tato práce se věnuje návrhu kmitočtových filtrů, především s použitím aktivních prvků, místo induktorů, které jsou na nízkých kmitočtech špatně realizovatelné. Zejména se zaměřuje na počítačové zpracování návrhu těchto obvodů.

V první části se práce zaměřuje na základní rozdělení kmitočtových filtrů, řeší se zde taktéž v jakých pracovních režimech mohou pracovat a v neposlední řadě se zabývá prozkoumání různých návrhových metod. Největší důraz je kladen na metodu úplné admitanční sítě a metodu autonomních obvodů, kterou se v poslední části této práce snaží uvést do praxe za pomoci programu *Mathematica*.

V této kapitole se práce seznamuje s prostředím Wolfram Mathematici a jejíma výhodami a něco málo z její historie.

Další část práce se zaměřuje na vysvětlení Modifikované metody uzlových napětí, která se bude při samotném návrhu využívat a taky pravidla, která se musí respektovat, při nahrazení tranzistoru, zdrojem proudu řízeného napětím.

V poslední části práce se snaží uvést v chod algoritmus, který pracuje právě na principu návrhu pomocí autonomních obvodů a modifikované metody uzlových napětí.

1 KMITOČTOVÉ FILTRY

Kmitočtové jsou lineární dvojbrany [4],[6]. Jejich hlavním úkolem je výběr (selekce) kmitočtových složek signálů. Filtry buď propouštějí některé kmitočtové složky signálu bez útlumu (propustné pásmo) nebo naopak potlačují některé složky (pásmo potlačení). Pro vyjádření těchto vlastností obvykle používáme modulovou (amplitudovou) charakteristiku.

Při průchodu signálu filtrem dochází k časovému zpoždění signálu, což vede k fázovému posuvu procházejících harmonických složek signálu. Tento jev vyjadřujeme ve fázové kmitočtové charakteristice.

Přenos má propustné pásmo kmitočtů, nepropustné pásmo kmitočtů a pásmo přechodu. Strmost filtru je přechod mezi propustným a nepropustným pásmem, záleží na řádu filtru a typu aproximace.

Strmost filtru, což je vlastně přechod mezi propustným a nepropustným pásmem, je dána řádem filtru a typem aproximace.

1.1 Základní druhy kmitočtových filtrů

Rozdělení podle přenášeného spektra [6]:

- **dolní propust** (DP nebo častěji LP - low-pass):

Propouští nižší kmitočty než je jeho mezní frekvence f_m a vysoké kmitočty potlačuje. Mezní kmitočet nastane tehdy, když modulová charakteristika klesne o - 3 dB. Jeho časté využití je v usměrňovačích a antialiasingových filtrech. Přenosová funkce pro DP druhého řádu je obecně ve tvaru

$$K(p) = \frac{a_0}{b_2 p^2 + b_1 p + b_0}. \quad (1.1)$$

- **horní propust** (HP stejně jako HP - high-pass):

Propouští vyšší kmitočty než mezní frekvence f_m a nízké kmitočty odfiltruje. Uplatnění najde třeba jako výhybka pro vysoké kmitočty při použití vícepásmových reprosoustav. Přenosová funkce pro HP druhého řádu je obecně ve tvaru

$$K(p) = \frac{a_2 p^2}{b_2 p^2 + b_1 p + b_0}. \quad (1.2)$$

- **pásmová propust** (PP nebo BP - band-pass):

propouští signál jen na určitém kmitočtovém pásmu a zbytek pásma potlačuje. Použití v přijímačích, pro selekci jen určitého pásma vysílače, které chceme na přijímači přijímat. Přenosová funkce pro filtr druhého řádu v obecném tvaru

$$K(p) = \frac{a_1 p}{b_2 p^2 + b_1 p + b_0}. \quad (1.3)$$

- **pásmová zádrž** (PZ nebo BR - band-reject):

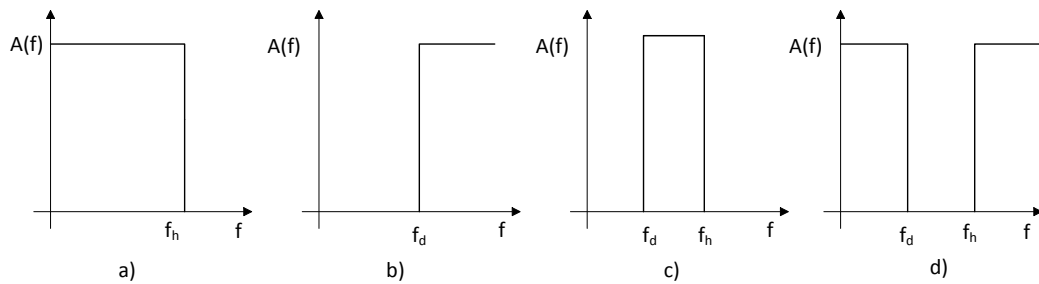
Potlačuje zvolené kmitočtové pásmo a zbylé frekvence propouští. Přenosová funkce pro filtr druhého řádu v obecném tvaru

$$K(p) = \frac{a_2 p^2 + a_0}{b_2 p^2 + b_1 p + b_0}. \quad (1.4)$$

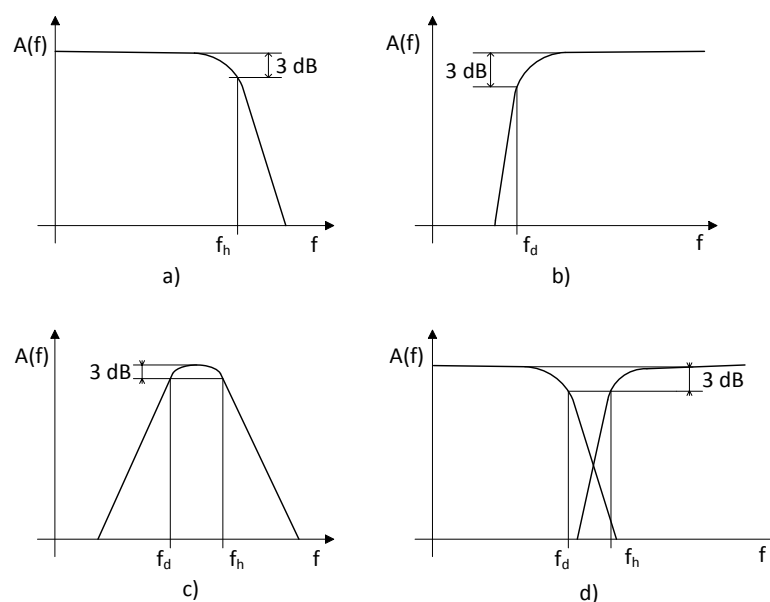
- **fázovací (všeprupustný) článek** (FČ nebo AP - all-pass):

Tyto obvody mají kmitočtově závislou fázovou charakteristiku, modulová je naopak nezávislá. Používají se jako zpožďovací články nebo pro korekci fázových charakteristik. Přenosová funkce pro filtr druhého řádu je v obecném tvaru

$$K(p) = \frac{p^2 - b_1 s + b_0}{p^2 + b_1 s + b_0}. \quad (1.5)$$



Obr. 1.1: ideální filtry a) DP b) HP c) PP d) PZ



Obr. 1.2: reálné filtry a) DP b) HP c) PP d) PZ

Rozdělení podle druhu použitých prvků:

- **pasivní:**

Jsou použity pouze pasivní součástky (rezistor, kapacitor, induktor). Pasivní RC filtry mají slabé selektivní vlastnosti. Používají se jen v těch aplikacích, kde není kladen takový nárok na přesnost přenosové funkce filtru. Když nahradíme rezistor cívkou, dostaneme strmější pasivní filtr RLC. Nahrazením takového filtru, filtrem ekvivalentním s aktivními prvky, dostaneme obvod se snazší přeladitelností parametrů.

- **aktivní:**

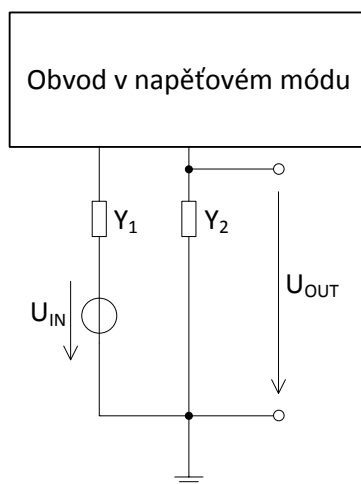
Aktivní filtr dostaneme, když pasivním prvkům, přidáme jeden, nebo i několik aktivních prvků (proudový nebo napěťový konvektor, operační zesilovač atd.). Jedna ze zásadních předností aktivních filtrů je možnost vyloučení induktorů z celého návrhu filtru i přenosové funkce.

1.2 Pracovní režim obvodu

Elektrické obvody pracují v různých režimech, záleží na tom s jakými signály pracují. V minulosti se využíval hlavně režim napěťový (snížená dynamika a SNR), v dnešní době se od něj upouští a snaží se přecházet na režim proudový nebo smíšený. V těchto režimech se dá dosáhnout lepších výsledků (snížení napájecí napětí, možnost použití na vyšších kmitočtech).

- **Napěťový mód (VM - Voltage Mode)**

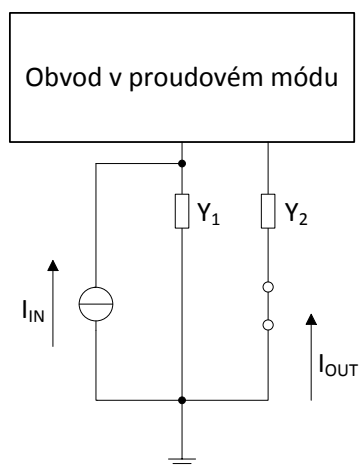
Obvod je buzen zdrojem napětí, který je připojen mezi pasivní dvojpól a zem, na výstupu je sledována napěťová odezva v uzlu vůči zemi (viz. 1.3).



Obr. 1.3: obvod v napěťovém módu

- **Proudový mód (CM - Current Mode)**

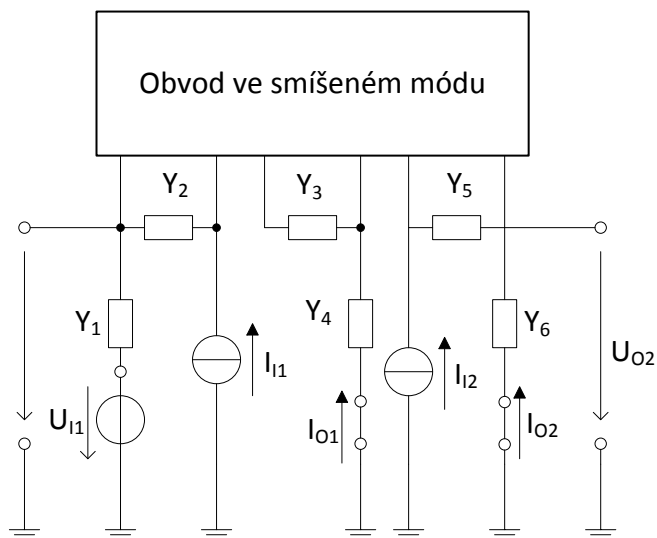
Obvod je buzen zdrojem proudu, připojen mezi uzel a zem, z výstupu je sledována proudová odezva ve větvi mezi dvojpólem a zemí (obr. 1.4). Tento mód má výhodu oproti napěťovému, že jde využít pro oblasti vysokých kmitočtů.



Obr. 1.4: obvod v proudovém módu

- **Smíšený mód**

Smíšený mód může pracovat buď v *napěťovém režimu* (V/CM), kdy při zpracování signálu je využito napěťové odezvy při proudovém buzení (výsledný obvod zpracovává proudový signál) nebo v *proudovém režimu* (C/VM), kdy je využito proudové odezvy při napěťovém buzení (při zpracování je využit napěťový signál).



Obr. 1.5: Obvod ve smíšeném módu

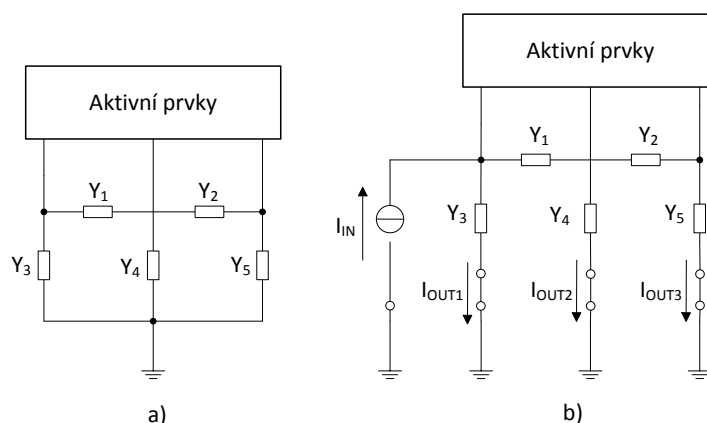
2 METODY NÁVRHU FILTRŮ S AKTIVNÍMI PRVKY

2.1 Návrh pomocí metody autonomních obvodů

Metoda autonomních obvodů [5], [11], [7] je jednou z nejpoužívanějších návrhových metod kmitočtových filtrů. Autonomní obvod je struktura pasivních a aktivních prvků, který neobsahuje žádné budící zdroje a nemá vyznačenou vstupní ani výstupní svorku viz obr. 2.1a. Jediné, co je na obvodu známo, je jeho charakteristická rovnice. Tato rovnice je determinantem admitanční matice analyzovaného obvodu a taky udává jakého je daný filtr řádu, to je dáno mocninou operátoru p . Rovnice musí obsahovat všechny členy Laplaceova operátoru a zároveň být kladné, aby byl filtr stabilní. Z této rovnice lze tedy vidět, že obvod se může chovat buď jako oscilátor nebo jako různé typy kmitočtových filtrů. Pro lepší realizovatelnost v integrovaných obvodech se snažíme, aby pasivní prvky byli pokud možno jedním koncem uzemněny. V autonomním obvodu se může vyskytovat libovolný počet aktivních prvků. Obr. 2.1b ukazuje autonomní obvod při připojení zdroje proudu do uzlu a snímání proudové odezvy ve všech větvích.

Jednotlivé kroky návrhu:

- Návrh obecného autonomního obvodu (na základě předchozích zkušeností nebo jen zkusmo), který se bude skládat ze zobecněných aktivních prvků a obecných dvojpólů charakterizovaných například admitancemi.
- Výpočet a zjednodušení charakteristické rovnice s ohledem na budoucí říditelnost filtru.
- Dosazení za pasivní dvojpóly (admitance se nahradí rezistory nebo kapacitory).
- Určení umístění vstupních a výstupních svorek v obvodu. Musí se respektovat to, že proudové zdroje se mohou připojovat pouze do uzlů obvodu a proudovou odezvu můžeme sledovat pouze ve větvích. Napěťové zdroje se smí připojovat pouze do větví a napěťová odezva se sleduje v uzlech.
- Určení přenosové funkce.
- Samotná realizace obvodu pomocí dostupného aktivního prvku.



Obr. 2.1: a) obecný autonomní obvod, b) proudový mód

2.2 Návrh pomocí úplné admitanční sítě

Metoda zobecňuje metodu autonomních obvodů, kdy nalezne všechna řešení autonomních obvodů. Prvním krokem je sestavení úplné admitanční sítě se zvoleným počtem aktivních prvků. Z ní se postupným zjednodušováním získá skupina autonomních obvodů s počtem pasivních prvků, který byl zvolen. Struktury, vhodné pro další zpracování, jsou ty, které obsahují co nejvíce uzemněných pasivních prvků. Kvůli snazší realizovatelnosti obvodu.

Pomocí této metody je možné najít všechny varianty řešení zapojení pro zvolený počet aktivních a pasivních prvků. Za pomoci dalších kritérií je možné najít z tohoto kvanta variant zapojení, přes to které bude odpovídat našim požadavkům.

Úplná admitanční síť [7] musí obsahovat $n(m+1)$ uzlů, kde n je počet aktivních prvků a m je počet bran aktivního prvku.

Nevýhodou této metody je velká časová náročnost, protože čím víc pasivních a aktivních prvků se nachází v obvodu, tím je řešení této metody zdlouhavější. Proto pro návrh kmitočtových filtrů vyššího řádu se častěji používá metoda rozšiřování již nalezených autonomních obvodů.

2.3 Rozšiřování autonomních obvodů

Pro návrhy filtrů vyššího řádu než druhého, je vhodnější použít již nalezené autonomní obvody a rozšířit je o daný počet pasivních a aktivních prvků, oproti návrhu pomocí úplné admitanční sítě, kde by se při velkém počtu pasivních a aktivních prvků řešení značně zkomplikovalo. Zvýšení řádu kmitočtového filtru je možné tehdy, když rozšíříme výchozí strukturu o další uzel nezávislého napětí.

Při použití proudových konvektorů [7], platí opatření, že při zapojení dalšího aktivního prvku do struktury, nesmí být napěťová brána Y zapojena do uzlu, do kterého je už připojena napěťová brána Y nebo proudová brána X některého z ostatních proudových konvektorů. Toto opatření značně napomáhá řešení návrhu kmitočtových filtrů vyššího řádu.

2.4 Syntéza filtrů pomocí grafů signálových toků

Grafy signálových toků [7], [8], [2] byly původně navrženy Masonem v roce 1953 pro popis a řešení lineárních obvodů. O několik let později se objevily zobecněné Coatesovy grafy. Graf je soustava bodů a úseček nazývaných jako uzly a větve. Každý konec větve je připojen k uzlu. Oba konce větve mohou být připojeny ke stejnému uzlu. Graf signálových toků je diagram, který znázorňuje vzájemný vztah mezi proměnnými. Tyto proměnné jsou reprezentovány uzly grafu a větve definují jejich vzájemný vztah.

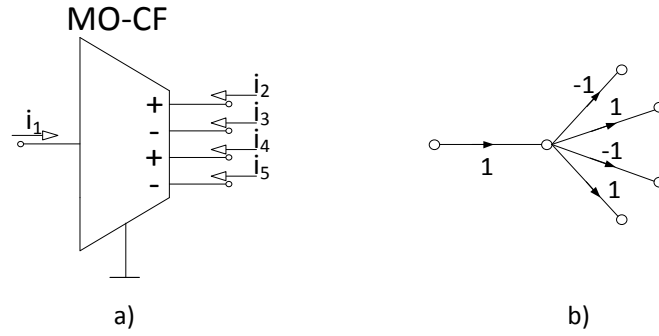
Pro účely analýzy a syntézy elektrických obvodů jsou používány smíšené Masonovy-Coatesovy grafy (M-C grafy) [1]. Tuto metodu je možno použít pro přímý návrh obvodů podle požadovaného tvaru přenosové funkce. Přenosovou funkci můžeme vypočítat pomocí Masonova pravidla

$$K = \frac{Y}{X} = \frac{1}{\Delta} \sum_i P_i \Delta_i, \quad (2.1)$$

kde P_i je přenos i -té přímé cesty ze vstupního uzlu X do výstupního Y a Δ je determinant grafu. Determinant je dán vztahem

$$\Delta = V - \sum_k S_1^{(k)} V_1^{(k)} + \sum_l S_2^{(l)} V_2^{(l)} + \sum_m S_3^{(m)} V_3^{(m)} + \dots, \quad (2.2)$$

kde V je součin všech neorientovaných vlastních smyček, $S_1^{(k)}$ je přenos k -té orientované smyčky a $V_1^{(k)}$ je součin všech vlastních smyček uzlů, kterých se k -tá smyčka nedotýká, $S_2^{(l)}$ je součin přenosů dvou nedotýkajících se smyček a $V_2^{(l)}$ je součin všech vlastních smyček uzlů, kterých se l -tá smyčka nedotýká. Pokud se smyčka nebo k -tá přímá cesta dotýká všech uzlů, pak součin V , resp. Δ_k , je identicky roven jedné. Δ_i je determinant části grafu, která se nedotýká i -té přímé cesty. Výsledek vzorce 2.2 představuje levou stranu charakteristické rovnice, která samozřejmě dominantně ovlivňuje chování daného obvodu. Můžeme tedy lehce navrhnout filtr s požadovanými parametry bez použití speciálního softwaru pro symbolickou analýzu. Navíc proces návrhu není intuitivní jako například u metody autonomních obvodů, protože u této metody existují přesně definovaná pravidla výše uvedená. Na obr.2.2 je původní a zjednodušené zobrazení aktivního prvku MO-CF.



Obr. 2.2: a) původní, b) zjednodušené zobrazení prvku MO-CF

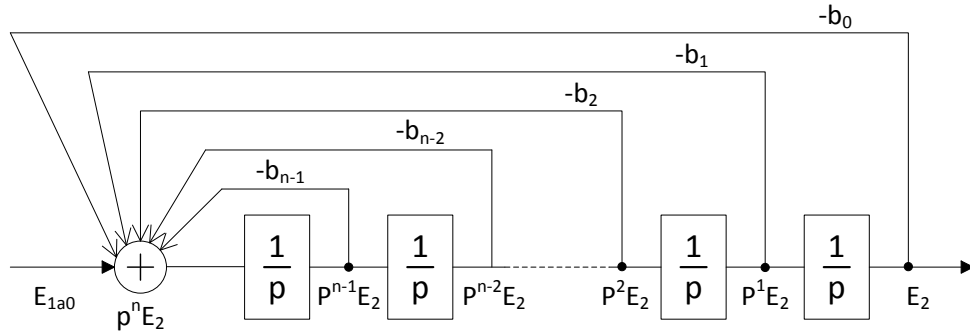
2.5 Návrh filtrů pomocí stavového popisu

Filtry navržené podle stavového popisu nazýváme také filtry s několikanásobnými zpětnými a dopřednými vazbami. Vyplynuly z požadavku získat přenosové funkce alespoň čtvrtého řádu. Tyto struktury nabízejí lepší citlivostní vlastnosti, než skýtá kaskádní syntéza. tavebním prvkem bývá invertující nebo neinvertující integrátor s přenosem $\pm 1/p$. Pomocí kaskádního řazení těchto prvků, jak ukazuje obr.2.3, dosáhneme příslušný typ přenosové funkce.

Kanonická struktura je taková struktura, která má minimální možný počet integrátorů pro daný řád filtru.

Přenosová funkce je v kanonickém tvaru

$$\frac{E_2}{E_1} = \frac{a_0}{p^n + p^{n-1}b_{n-1} + p^{n-2}b_{n-2} + \dots + p^2b_2 + p^1b_1 + b_0}. \quad (2.3)$$



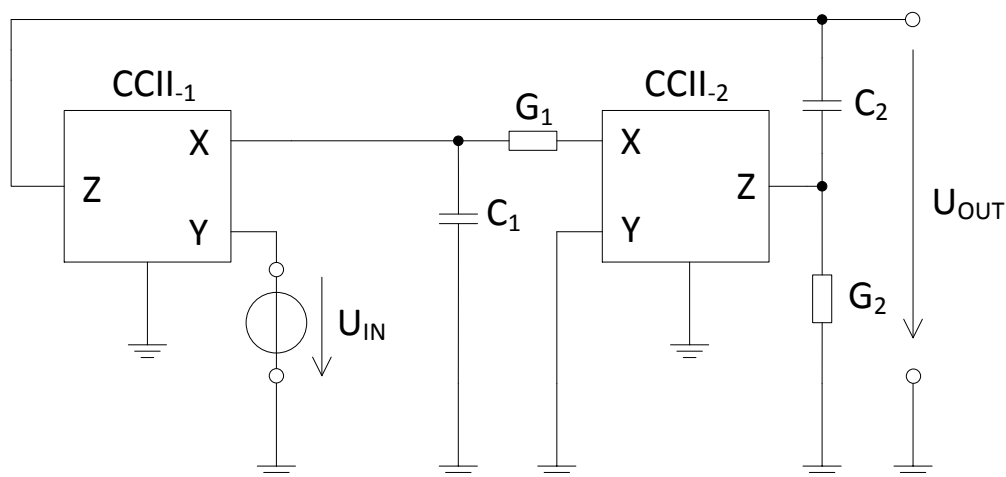
Obr. 2.3: Kaskádní zapojení neinvertujících integračních článků se zpětnými vazbami

2.6 Přidružená transformace

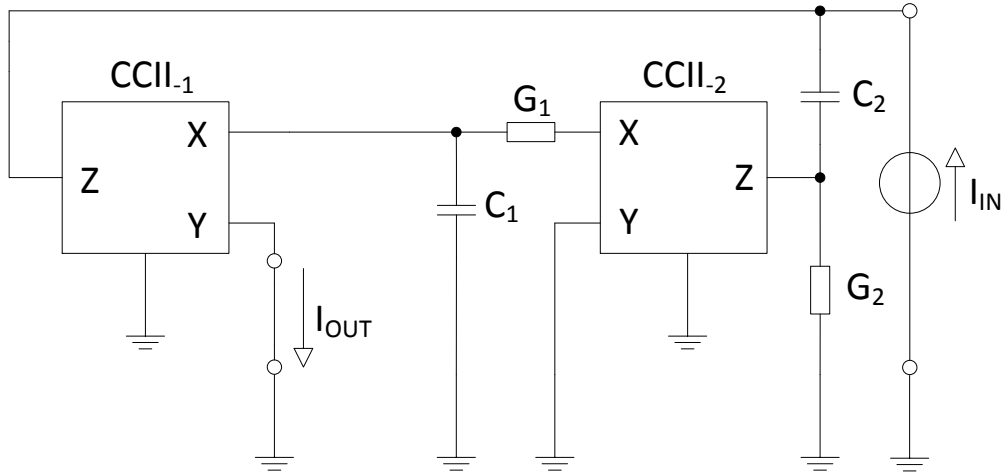
Je výhodné, když obvody pracují v proudovém módu. Abychom mohli získat proudový mód, využijeme napěťový mód a pomocí přidružené transformace získáme proudový mód.

Přidružená transformace [3] nám poskytuje přímý a jednoznačný přechod mezi napěťovým a proudovým módem. Často se obvody, pracující v napěťovém módu, převádějí na proudový mód a naopak.

V publikaci [3] jsou mimo jiné uvedeny tabulky s přidruženými prvky v proudovém a napěťovém módu. V publikaci [10] je uveden filtr typu invertující pásmová propust s proudovými konvejory CCII- v napěťovém módu obr. 2.4. Využijeme přidružené transformace k získání stejného typu filtru pracujícího v proudovém módu. Přidružený (adjoint) obvod tedy získáme z původního obvodu tak, že se změni buzení z napěťového na proudové. Zároveň se musí v tomto případě zaměnit vstupní a výstupní brána filtru. Čili budicí napěťový zdroj bude nahrazen zkratem, přes který pak v přidruženém obvodu teče výstupní proud. Budicí proudový zdroj bude připojen na původní výstupní bránu aktivního prvku, jak ukazuje obr. 2.5. Obvody na obr. 2.4 a obr. 2.5 jsou zde uvedeny pouze jako příklad.



Obr. 2.4: Obvod pracující v napěťovém módu



Obr. 2.5: Ukázka aplikace přidružené transformace, obvod pracující v proudovém módu

2.7 Využití syntetických prvků s imitancí vyšších řádů pro návrh kmitočtových filtrů

Původně se tato teorie věnovala návrhu klasických induktorů [7], které jsou ovšem na nízkých kmitočtech rozměrné a drahé. Na vysokých kmitočtech je tento problém odstraněn pouze v tom případě, je-li induktor realizován v diskrétní podobě. Pro realizaci obvodové struktury v integrované podobě, je nutné induktor nahradit syntetickým prvkem.

Při návrhu nových funkčních bloků je možné definovat syntetické prvky vyšších řádů [12], které při zapojení s kmitočtově závislými děliči napětí nebo proudu, lze realizovat filtry námi požadovaného řádu. Nemusíme se tedy omezovat na realizaci syntetických induktorů nebo kmitočtově závislých negativních rezistorů (FNDR).

Syntetické prvky s imitancemi vyšších řádů jsou tvořeny sériovým nebo paralelním zapojením elementárních dvojpólů typu D nebo E řádu (viz tab. 2.1).

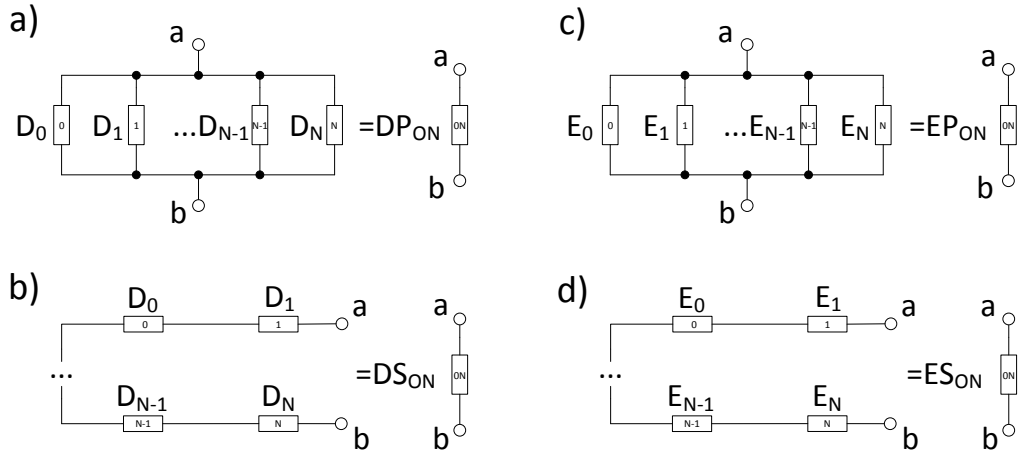
Tab. 2.1: Souhrn elementárních dvojpólů

typ	řád	název	imitance	typ	řád	název	imitance
E	0.	rezistor	$Z(j\omega) = R$	E	2.	„dvojný“ induktor	$Z(j\omega) = -\omega^2 E_2$
D	0.	konduktor	$Y(j\omega) = G$	D	2.	„dvojný“ kapacitor	$Y(j\omega) = -\omega^2 D_2$
E	1.	induktor	$Z(j\omega) = j\omega L$	E	N_E	obecný induktor	$Z(j\omega) = (j\omega)^{N_E} E_{N_E}$
D	1.	kapacitor	$Y(j\omega) = j\omega C$	D	N_D	obecný kapacitor	$Y(j\omega) = (j\omega)^{N_D} D_{N_D}$

Podle [12] lze definovat čtyři zapojení soustavy syntetických elementárních dvojpólů s imitancemi vyšších řádů:

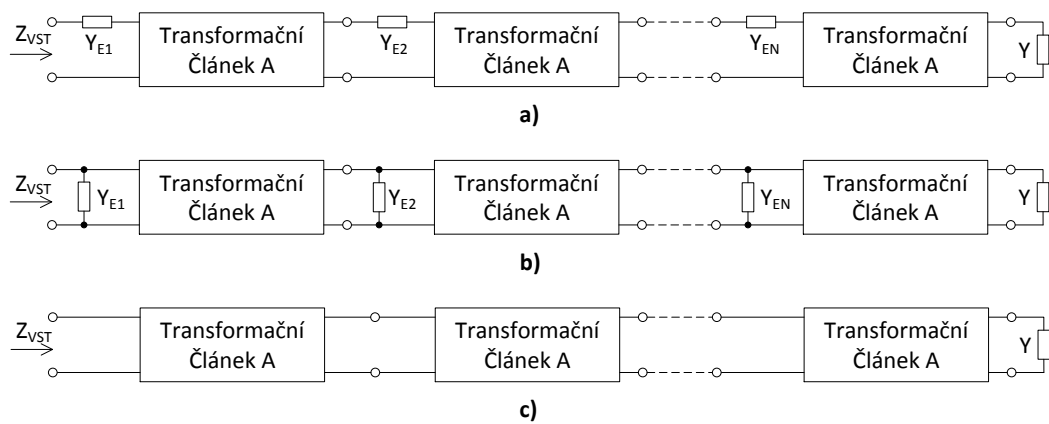
- dvojpól označený DP $N_{D,\min} N_{D,\max}$ (obr. 2.6a) vzniklý paralelním spojením syntetických elementárních dvojpólů typu D_n pro $n = N_{D,\min}, N_{D,\min} + 1, \dots, N_{D,\max} - 1, N_{D,\max}$,
- dvojpól označený DP $N_{D,\min} N_{D,\max}$ (obr. 2.6b) vzniklý sériovým spojením syntetických elementárních dvojpólů typu D_n pro $n = N_{D,\min}, N_{D,\min} + 1, \dots, N_{D,\max} - 1, N_{D,\max}$,
- dvojpól označený EP $N_{E,\min} N_{E,\max}$ (obr. 2.6c) vzniklý paralelním spojením syntetických elementárních dvojpólů typu E_n pro $n = N_{E,\min}, N_{E,\min} + 1, \dots, N_{E,\max} - 1, N_{E,\max}$,
- dvojpól označený ES $N_{E,\min} N_{E,\max}$ (obr. 2.6d) vzniklý sériovým spojením syntetických elementárních dvojpólů typu E_n pro $n = N_{E,\min}, N_{E,\min} + 1, \dots, N_{E,\max} - 1, N_{E,\max}$,

ty lze rozdělit na plovoucí a uzemněné.



Obr. 2.6: Syntetické dvojpóly typu a), DP, b) EP, c) DS, d) ES [12]

Výsledný kmitočtový filtr je stabilní tehdy, když použitý syntetický prvek bude obsahovat všechny řády elementárních dvojpólů od nejnižšího řádu (tj. 0. nebo 1. řád) po nejvyšší řád (tj. N-tého řádu). Pro realizaci je nutno použít soustavu transformačních článků. Každý článek navyšuje řád imitance na své výstupní bráně (obr. 2.7). V případě obr. 2.7a,b jsou použity vnější pasivní prvky a tím se navýší i celkový počet pasivních prvků v obvodu, oproti případu na obr. 2.7c.



Obr. 2.7: Obecná realizace syntetických prvků vyšších řádů a) sériové připojení vnější admitance b) paralelní připojení vnější admitance c) bez nutnosti připojení vnější admitance [7]

Strukturou s transformačními články A, která je zobrazena na obr. 2.7a lze realizovat imitanci prvky typu DS a ES (obr. 2.6d,c). Struktura na obr. 2.7b je vhodná pro realizaci syntetických prvků DP a EP (obr. 2.6a,b).

3 WOLFRAM MATHEMATICA

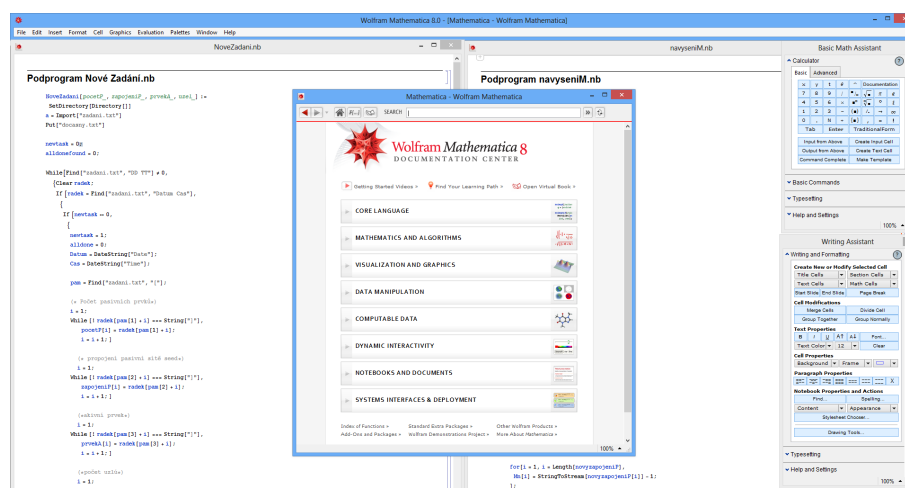
3.1 Využití, historie

Software Mathematica společnosti Wolfram Research byl poprvé uvedený na světlo světa dne 23. června 1988. Jeho autorem je britský matematik Stephen Wolfram. V dnešní době je jedním z nejlepších výpočetních systémů, ne-li nejlepší. Hlavním důvodem, který stál vznikem Mathematici, bylo vytvořit samostatný systém, který by dokázal zvládnout všechny hlediska technického programování. Toho je dosaženo vytvořením nového typu symbolického počítačového jazyka využívající prvky umělé inteligence. [14]

V dnešní době je Mathematica používána v matematice, fyzice, geografii, ekonomii, biologii, chemii, nebo informatice. Je používána při výuce na středních školách a univerzitách po celém světě nebo na všech 15 ministerstvech americké vlády. [14]

Mathematica pracuje s daty jakéhokoli typu, např. s chemickými daty, geografickými, socioekonomickými, finančními nebo astronomickými. Dále je možné automaticky importovat nebo exportovat data v jednom z více než 200 podporovaných formátů, připojovat se k externím programům, databázím nebo webovým stránkám, případně odesílat data přímo z programu na e-mail.

V Mathematice lze provádět okamžité výpočty a vizualizace, vytvářet záznamy práce, vytvářet interaktivní dokumenty a prezentace, přistupovat přímo do hlavní databáze nebo prostřednictvím Internetu ke vzdáleným databázím, vytvářet interaktivní dokumenty a prezentace, psát programy v jazyce Mathematica nebo vytvářet rozsáhlé aplikace. [14]



Obr. 3.1: prostředí Mathematici

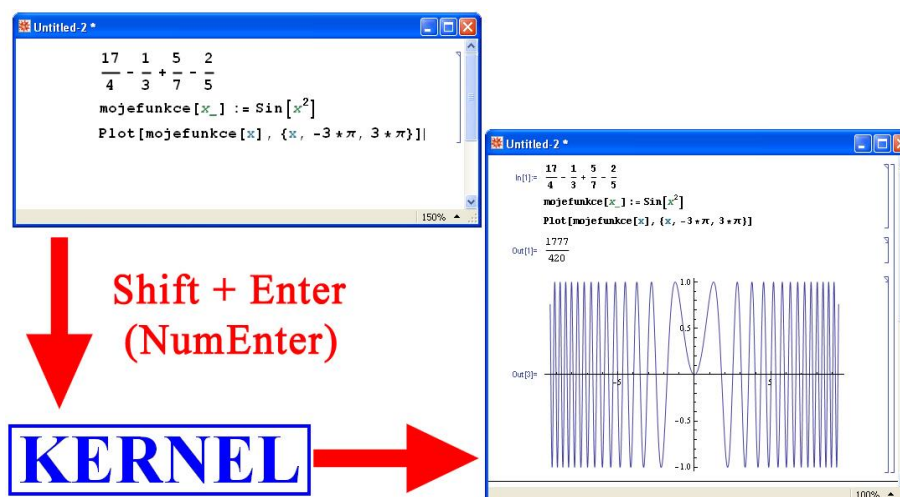
3.2 Prostředí Mathematici

Při práci s Mathematicou se používá pro zápis soubor, který se nazývá notebook, který všechno co děj napíšeme organizuje do spustitelného a interaktivního dokumentu. Základním stavebním prvkem notebooku je buňka (Cell), do níž se píše veškeré příkazy. Pro každý příkaz je v notebooku vstupní a výstupní buňka.

Požadavek na výpočet buňky se předává jádru, tzv. Kernel, který požadavek vyhodnotí a vykoná. Jako vstupní proměnná může být použit kód, text, grafika, matematický vzorec, zvuk, animace nebo video.

Mathematica může komunikovat s jinými technickými programy jako MATLAB, LabView a jinými pomocí svého rozhraní a také s pomocí protokolu MathLink umožňuje volat funkce i z jiných programovacích jazyků jako, C, .NET, Java, Haskell nebo Visual Basic. Podporuje také paralelní výpočty.

Mathematica je dostupná téměř pro všechny komerčně používané platformy operačních systémů Windows, Linux, Mac OS a Solaris, ve verzích pro 32bitové a 64bitové architektury procesorů.



Obr. 3.2: ukázka práce Kernelu [13]

4 AUTOMATIZACE NÁVRHU LINEÁRNÍCH OBVODŮ

4.1 Princip modifikované metody uzlových napětí

Metoda vychází z klasické metody uzlových napětí[9], která se často používá pro „počítačové“ řešení lineárních obvodů a je vyjádřena rovnicí

$$\mathbf{Y} \cdot \mathbf{U} = \mathbf{I}, \quad (4.1)$$

kde \mathbf{Y} je admitanční matice soustavy, \mathbf{U} vektor neznámých uzlových napětí a \mathbf{I} je vektorem všech budících proudů do jednotlivých uzlů soustavy.

Nevýhodou této metody je, že nelze řešit obvody která obsahují řízené zdroje, až na výjimku (napětím řízený zdroj proudu) a taky nemožnost řešením určit proudy jednotlivých větví. Tyto nedostatky právě řeší modifikovaná metoda uzlových napětí (MMUN).

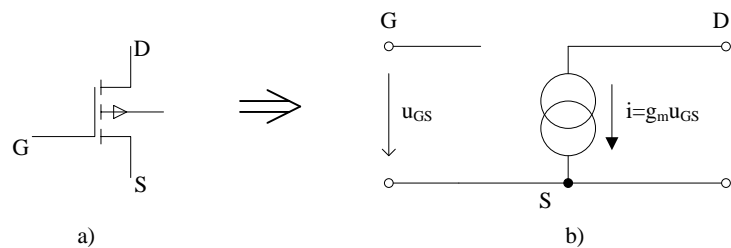
Samotný princip metody vyplývá z toho, že proměnnou nemusí být jen napětí, ale i proud nebo některé z pravých stran rovnic obsahuje princialně 0 (tzn. nelze budit do daného uzlu - u napětových zdrojů). Tedy matice prvků nelze nazývat admitanční ani vodivostní, ale lze pomocí této matice popsat téměř jakýkoliv lineární prvek a obvod daný rovnicí

$$x(u_a - u_b) + yi_k = z, \quad (4.2)$$

kde u_a a u_b jsou napětí na prvku, i_k proud daný prvkem a x, y, z jsou konstanty.

4.2 Tranzistor MOSFET a jeho nahrazení zdrojem proudu řízeného napětím (VCCS)

Tranzistor MOSFET, lze pro jednodušší algoritmizaci převést na jeho náhradní model pomocí zdroje proudu řízeného napětím (obr.4.1). Pro zařazení VCCS do obvodu musíme k původní matici přičíst v odpovídajících uzlech matici následující, jedná se o tzv. „razítko“ (viz tab.4.1).



Obr. 4.1: a) tranzistor MOSFET b) náhradní model tranzistoru pomocí VCCS

Tab. 4.1: razítko VCCS

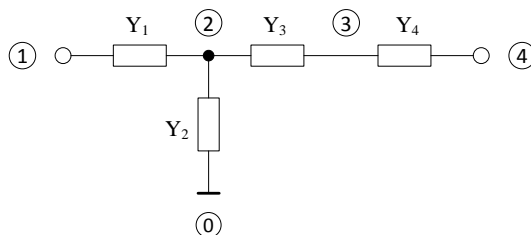
G	S	D
$-g_m$	g_m	
g_m	$-g_m$	

4.3 Popis automatizace návrhu autonomního obvodu

Automatizace návrhu spočívá v tom, že je vyvinut algoritmus na základě definice možnosti spojení mezi pasivními prvky (které mohou a nemusí být uzemněny) a mezi prvky aktivními. Příklad zapojení pasivní sítě i s popisem pro „počítačové“ zpracování (proměnná `code-p`) je na obrázku 4.2. Připojení aktivních prvků do korespondujících bodů pasivní sítě je popsáno v proměnné `code-a`. Celý autonomní obvod je popsán v proměnné `code`. Proměnná `code-p` je v podstatě admitanční matice Y , zápis této proměnné se tedy dá popsat následující maticí

$$Y = \begin{bmatrix} 1 & 2 \\ 2 & 0 \\ 2 & 3 \\ 3 & 4 \end{bmatrix}. \quad (4.3)$$

První řádek matice vyjadřuje prvek Y_1 , druhý řádek Y_2 a tak dále.



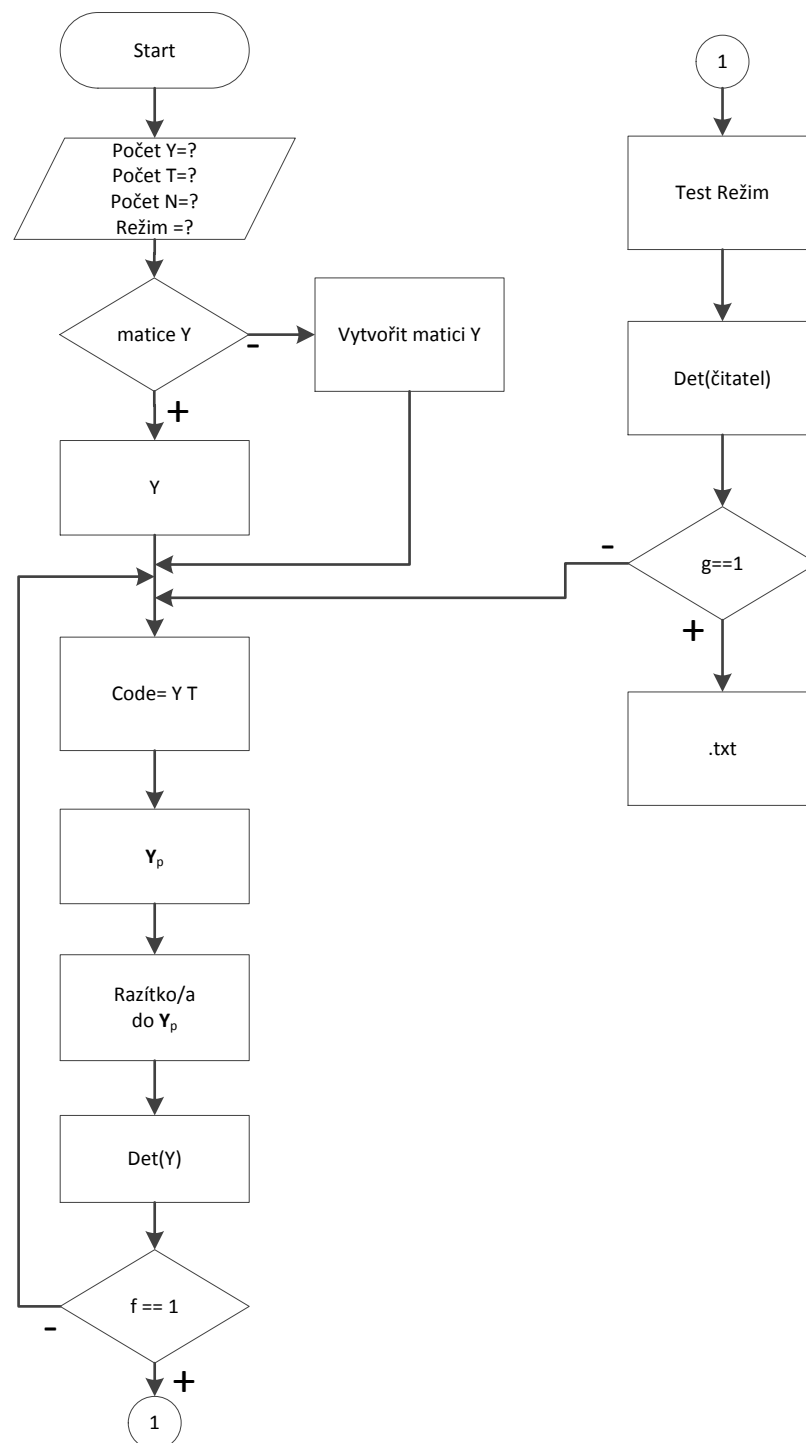
Obr. 4.2: Př. zapojení pasivní sítě (`code-p` = 12202334)

4.4 Teoretické řešení realizace automatizovaného návrhu

Na začátku programu je třeba definovat počet admitancí Y_x a taky počet aktivních prvků T_x (v našem případě budou aktivními prvky tranzistory typu MOSFET, ale vyjádřeny pomocí náhradního obvodu – zdroje proudu řízeného napětím s parametrem g_m). Dále bude potřeba určit v jakém **pracovním režimu** bude obvod pracovat (VM, CM nebo smíšeném) a taky bude možnost zvolit maximální počet uzlů N . Když máme nadefinované základní parametry budoucího obvodu, můžeme vytvořit admitanční matici.

Jakmile máme admitanční matici, postupně připojujeme aktivní prvky a hledáme jedinečné kombinace jejich spojení `code = Y T`. V dalším bodě admitancím přidáme reálný charakter, přidáním Laplaceova operátoru p a v příslušných uzlech přičteme k původní matici razítko tranzistorů T_x .

Pokud se našel autonomní obvod $f = 1$, začneme testovat režim obvodu, pokud byl v čitateli nalezen alespoň jeden člen $g=1$, ve jmenovateli máme všechny prvky operátoru p a jsou všechny kladné nebo záporné, dostáváme stabilní filtr. Jedinečnou kombinaci spojení prvků a přenosovou funkci, uložíme do externího souboru např. `.txt`.



Obr. 4.3: vývojový diagram programu

5 PROGRAMOVÉ ŘEŠENÍ

Celé programové řešení se skládá ze tří dílčích částí, a to z: *HlavniProgram.nb*, *NoveZadani.nb* a *NavyseniM.nb*. Dále jako soubor se vstupními hodnotami obvodu je použit textový soubor *zadani.txt*.

```
(*      4 uzemnene      *)
Datum Cas pocetP: [4] zapojeniP: [01020304] prvekA: [VCCS] uzel: [5]
(*      3 uzemnene      *)
Datum Cas pocetP: [4] zapojeniP: [01020323] prvekA: [VCCS] uzel: [4]
Datum Cas pocetP: [4] zapojeniP: [01020334] prvekA: [VCCS] uzel: [5]
Datum Cas pocetP: [4] zapojeniP: [01020345] prvekA: [VCCS] uzel: [6]
(*      2 uzemnene      *)
Datum Cas pocetP: [4] zapojeniP: [01021324] prvekA: [VCCS] uzel: [5]
Datum Cas pocetP: [4] zapojeniP: [01021213] prvekA: [VCCS] uzel: [5]
Datum Cas pocetP: [4] zapojeniP: [01021345] prvekA: [VCCS] uzel: [6]
```

Obr. 5.1: část vstupního souboru *zadani.txt*

Řetězce *Datum* a *Cas*, slouží pro nalezení jednotlivých řádků. Proměnná *pocetP* značí počet pasivních prvků v obvodu (v našem případě Kondenzátorů). Další proměnnou je *zapojeniP*, v které je uložena kombinace zapojení pasivních prvků, která je poskládá tak jako je to uvedeno v předcházející kapitole. V proměnné *prvekA* je uloženo zapojení aktivního prvku VCCS, které bude popsáno dále. Jako poslední proměnnou je *uzel*, ve které je počet uzlů pasivních prvků.

5.1 Podprogram - NoveZadani.nb

- Na začátku je nadefinovaná funkce pojmenována *NoveZadani*, která se vytváří na základě parametrů pro počet pasivních prvků, jejich zapojení, počtu uzlů a aktivního prvku VCCS. Dále je nastavený aktuální pracovní adresář a je nainportován soubor se vstupními hodnotami *zadani.txt* a vytvořený soubor *docasny.txt*. Proměnné *noveZadani* a *hotovoNalezen* nastaveny na nulu.

```
NoveZadani[pocetP_, zapojeniP_, prvekA_, uzel_] :=
  SetDirectory[Directory[]]
a = Import["zadani.txt"]
Put["docasny.txt"]

noveZadani = 0;
hotovoNalezen = 0;
```

- Find prohledává zadani.txt, než najde řetězec "Datum Cas", to se děje do té doby než narazí na konec souboru. Když nalezne řetězec, tak pokračuje dál v programu. Nastaví hodnotu noveZadani na jedna a hotovo na nula. Do proměnný Datum a Cas uloží aktuální datum a čas.

```
While[Find["zadani.txt", "Datum Cas"] ≠ 0,
{Clear radek;
If [radek = Find["zadani.txt", "Datum Cas"],
{
If [noveZadani == 0,
{
noveZadani = 1;
hotovo = 0;
Datum = DateString["Date"];
Cas = DateString["Time"];

```

- Když nalezne řetězec "VCCS" přidá vývody aktivního prvku k zapojeniP a za Datum a Cas nastaví skutečný aktuální datum a čas. Uloží všechny provedené změny do souboru docasny.txt

```
vyvody = 0;
If[prvekA = Find[a, "VCCS"],
vyvody = vyvody + 3]

zapojeniP[Length[zapojeniP] + vyvody] = char["0"];

radek = StringReplace["Datum", Datum];
radek = StringReplace["Cas", Cas];

radek >> "docasny.txt";
}
,
radek >> "docasny.txt";
```

- Jakmile je nalezen v souboru řetězec "hotovo", nastaví se hotovo i hotovoNalezen na jedna, do ostatních proměnných se přiřadí znak "-".
Když není nalezeno noveZadani ani hotovoNalezen, tak se připiše do souboru docasny.txt slovo „hotovo“, proměnná hotovo se nastaví na jedna a do ostatních proměnných se přiřadí znak "-".
Nalezeno noveZadani, vytvoří se soubor ukol.txt, do něj se uloží proměnné pocetP, zapojeniP, prvekA, uzel a hotovo.

```

If[ radek = Find["zadani.txt", "hotovo"],
    {hotovoNalezen = 1;
     hotovo = 1;},
    {radek >> "doca.txt";
     hotovoNalezen = 0
    }];
If [noveZadani == 0 && hotovoNalezen == 0,
    {"hotovo" >>> "docasny.txt";
     hotovo = 1;
     pocetP = "-";
    }];
If[noveZadani == 1,
    {Put["ukol.txt"];
     pocetP >>> "ukol.txt";
     zapojeniP >>> "ukol.txt";
     prvekA >>> "ukol.txt";
     uzel >>> "ukol.txt";
     ToString[hotovo] >>> "ukol.txt"
    }];

```

5.2 Hlavní program - HlavniProgram.nb

- Nastaví se pracovní adresář. Zjistí se zdali již existuje `ukol.txt`, pokud ano, použije se funkce `NoveZadani` s parametrami `pocetP`, `zapojeniP`, `prvekA` a `uzel`, pokud není nalezen, otevře se pro čtení a najdou se v něm řetězce `pocetP`, `zapojeniP`, `prvekA` a `uzel`.

```

SetDirectory[Directory[]]
If [FileExistsQ["ukol.txt"],
    {NoveZadani[pocetP_, zapojeniP_, prvekA_, uzel_]
    },
    ukol = OpenRead["ukol.txt"],
    Find[ukol, "pocetP"];
    Find[ukol, "zapojeniP"];
    Find[ukol, "prvekA"];
    Find[ukol, "uzel"];
    ToString[hotovo] >> "ukol.txt";];

```

- Byl vytvořen vektor `M` z pasivních prvků `zapojeniP`, včetně referenčního uzlu 0. Když bude nějaký vývod aktivního prvku připojen do toho uzlu, tak se s ním nebude počítat. Dále byla vytvořena matice pasivních prvků `Yp`, viz popis u kódu.


```

For[i = 1, i = Length[zapojeniP], i++, M[i] = StringToStream[zapojeniP[i]] + 1];
Clear[Y, Yp, i]
For[i = 1, i = pocetP, i++, Y[i] = Symbol[ToString[i]]];
uzel = StringToStream[uzel];
Yp = Symbol[uzel + 1];
For[i = 0, i = pocetP, i++,
{
  Yp[M[2 * i - 1], M[2 * i - 1]] = Yp[M[2 * i - 1], M[2 * i - 1]] + Y[i];
  (*prvky připojeny k zemi*)
  Yp[M[2 * i], M[2 * i]] = Yp[M[2 * i], M[2 * i]] + Y[i]; (*prvky připojeny k zemi*)
  Yp[M[2 * i - 1], M[2 * i]] = Yp[M[2 * i - 1], M[2 * i]] - Y[i];
  (*prvky připojeny mezi uzly*)
  Yp[M[2 * i], M[2 * i - 1]] = Yp[M[2 * i], M[2 * i - 1]] - Y[i]
  (*prvky připojeny mezi uzly*)
}];

```

- Proměnná `nalezen` je nastaven na nule. Pokud je některý z uzlů ve vektoru `M` vyšší, než `uzel+1` nebo je roven nule, tak se volá funkce `navyseniM`.
Dál se testuje zda-li aktivní prvek je připojen k pasivní síti v dostatečném počtu uzlů (nejméně ovšem `vyvody/2`).

```

analpropojeni = 1;
If[analpropojeni == 1,
  vyvody = 0;
  For[i = 0, i = Length[prvkyA], i++,
    {If[Find[prvkyA[i], "VCCS"],
      vyvody = vyvody + 3;];};];
aktivuzel = Unique[M[2 * pocetP + 1 : Length[M]]];
relevantuzel = [];
j = 1;
For[i = 1, i = Length[aktivuzel], i++,
  {If[aktivuzel[i] ≤ Max[M[1 : 2 * pocetP]],
    {relevantuzel[j] = aktivuzel[i];
     j++;};};];];
If[Length[relevantuzels] ≥ vyvody / 2,
  analpropojeni = 1,
  analpropojeni = 0];]

```

- Obdoba předchozí analýzy, jen se netestuje připojení aktivní a pasivní sítě, ale pasivní prvků s pasivními prvkama. Nejméně $\text{Max}[M[1 : 2 * \text{pocetP}]] - 1$ uzlů musí být zapojeno v pasivní síti, aby nedocházelo v autonomních obvodech k seriovým kombinacím pasivních prvků. Nezajímá nás referenční uzel `pasuzel[1]`. Určíme rozdíl pasivních a aktivních uzlů, rozdílem zůstanou jen nepoužité aktivní uzly. Když bude délka `delta` větší než 1, tak je v pasivní síti víc, než jeden nevyužitý uzel.

```

If[analpropojeni == 1,
  {pasuzel = {};
   pasuzel = Unique[M[1 : 2 * pocetP]];
   If[pasuzel[1] == 1,
     {pasuzel[1] = {}
    }];
   If[aktivuzel[1] == 1,
     {aktivuzel[1] = {}
    }];
   delta = {};
   delta = DifferenceDelta[pasuzel, aktivuzel];
   If[Length[delta] > 1,
     analpropojeni = 0;
     , analpropojeni = 1
   ];}]];

```

- Definování matice pro náhradní schéma MOSFET tranzistoru, přidání parametru gm. Vypočítání determinantu.

```

vyvody = 0;
Ya = Yp;
For[i = 0, i = Length[prvkyA], i++,
  {
    If[prvkyA[i] == "VCCS",
      {
        vyvody = vyvody + 3;
        Ya[M[2 * pocetP + vyvody - 2], Max[M]] =
          Ya[M[2 * pocetP + vyvody - 2], Max[M] + gm]; (* G - M[2*pocetP-2] *)
        Ya[M[2 * pocetP + vyvody - 1], Max[M]] = Ya[M[2 * pocetP + vyvody - 1], Max[M] + gm];
        (* D - M[2*pocetP-2] *)
        Ya[M[2 * pocetP + vyvody], Max[M]] = Ya[M[2 * pocetP + vyvody], Max[M] - gm]
        (* S - M[2*pocetP-2] *)
      }];}]];
detY = Det[Ya];
detY = Expand[detY];
detYchar = ToCharacterCode[detY];

```

- Test, zda-li není v determinantu mínus. Poté se determinant rozdělí na části detcast[i] a provede se série dalších testů.

Test1: Když nebudou všechny admitance v jedné části determinantu, pak součin ve všech sloupcích YID musí být 0.

Test2: Součet všech sloupců musí být větší jak 0, když vyjde determinant.

Test3: Jakmile je v determinantu součin 2 a více prvků, tak součet řádků v YID musí být větší než 1.

Po nalezení, vyhovujícího obvodu, je vytvořen soubor `vysledky.txt` a do něho zapsány proměnné `pocetP`, `M`, `prvekA`, `detYchar` a je zavolán podprogram `navyseniM.nb`

```
(*rozdeleni na casti*)
If[minusAno == 0 && Length[zn] ≤ Ceiling[pocetP / 2 + 1]
  && Length[zn] ≥ Floor[pocetP / 2 + 1],
{detcast = {}];
For[i = 1, i = Length[zn] - 1, i++,
  detcast[i] = detYcharN[zn[i] + 1 : zn[i + 1] - 1];];
YID = ConstantArray[0, Length[detcast], pocetP];
detcastChar = ToCharacterCode[detcast];
For[i = 1, i = Length[detcast], i++,
  {pos = Find[detcast[i], "Y"]];
  For[j = 1, j = Length[pos], j++,
    YID[i, StringToStream[detcastChar[i, pos[j] + 1]]] = 1;
  ]; Clear[pos];]];

(*Test 1*)
For[i = 1, i = pocetP, i++,
  YIDSouc[i] = Product[YID, {YID[All, i]}];]
If[Sum[YID, i] == 0,
  nalezen = 1;]

(*Test 2*)
If[nalezen == 1,
  {For[i = 1, i = pocetP, i++,
    YIDSoucet[i] = Sum[YID, {All, i}];]
  If[Min[YIDSoucet] ≠ 0,
    nalezen = 1;,
    nalezen = 0;];]];

(*Test 3*)
If[nalezen == 1,
  {If[Sum[YID, {1, All}] > 1,
    nalezen = 1;,
    nalezen = 0;
  ];}]
```

5.3 Podprogram - navyseniM.nb

- Vytvořena funkce `navyseniM` s parametrami `M` a `t`. Z proměnné vektoru `M`, bere posledních 7 pozic, převede ho na stream a přičte k němu 1. Pokud je `t` větší než 1000 a zároveň `Min[M]` je větší jak 1, tak se provede kopie souboru `ukol.txt`

do souboru `copyukol.txt`. Soubor `ukol.txt` se otevře pro čtení a načtou se z něj proměnné `pocetP`, `zapojeniP`, `prvekA`, `uzel`, `hotovo`.

Ze zapojení proměnné `Mse` vytvoří `ZapojeniPn`, kde je uzemnění, reprezentována 1, z tohoto zapojení je vytvořeno nové zapojení `Mn`, kde je uzemnění reprezentováno 0, `Mn` je opět převedeno na string a jako proměnná `noveZapojeniP` je uloženo do souboru `ukol.txt`, s dalšíma proměnnýma `pocetP`, `prvekA`, `uzel` a `hotovo`.

```
navyseniM[M_, t_] :=
  For[i = 1, i = 7, i++,
    a[i] = {M[7 + i]};];
b = StringToStream[a[1 : 7]];
b = b + 1;
b = ToString[b];
For[i = 1, i = length[b], i++,
  M[7 + i] = ToString[b[i]];];
If [t ≥ 1000 & Min[M] ≥ 1,
  {If[FileExistsQ["copyukol.txt"],
    DeleteFile["copyukol.txt"]];
  CopyFile["ukol.txt", "copyukol.txt"];
  ukol = OpenRead["ukol.txt"];
  While[Read["ukol.txt"] ≠ EndOfFile,
    {Find["ukol", "pocetP"];
     Find["ukol", "zapojeniP"];
     Find["ukol", "prvekA"];
     Find["ukol", "uzel"];
     Find["ukol", "hotovo"];
     StringToStream[hotovo];}];
  Close[ukol];
  DeleteFile["ukol.txt"];
  For[i = 1, i = Length[M],
    zapojeniPn[i] = ToString[M[i]];];
  for[i = 1, i = Length[novezapojeniP],
    Mn[i] = StringToStream[novezapojeniP[i]] - 1;];
  for[i = 1, i = Length[Mn],
    novezapojeniP[i] = ToString[Mn[i]];];
  pocetP >>> "ukol.txt";
  novezapojeniP >>> "ukol.txt";
  prvekA >>> "ukol.txt";
  uzel >>> "ukol.txt";
  ToString[hotovo] >>> "ukol.txt";};
```

6 ZÁVĚR

Ve své bakalářské práci jsem, zabýval vývojem softwaru pro automatizovaný návrh kmitočtových filtrů, za pomoci vývojového prostředí Wolfram Mathematica. Pro vytvoření tohoto programu bylo první potřeba nastudovat teorii, týkající se metod návrhu s aktivními prvky a to hlavně metodu návrhu za pomoci autonomních obvodů.

V první části práce, byly probrány základní typy kmitočtových filtrů a pracovní režimy obvodu, ve kterých se mohou nacházet.

Ve druhé části, jsou stručně probrány jednotlivé metody návrhu filtrů, jako již zmíněná metoda autonomních obvodů. Z dalších například syntéza filtrů pomocí grafů signálových toků, tzv. M-C grafy nebo metoda pomocí úplné admitanční sítě.

Další část je věnovaná představení prostředí Wolfram Mathematici, její historii a kde všude se dnes využívá jejich možností a hlavně jaké jsou její přednosti.

Čtvrtá kapitola se věnuje teoretické popsání, co se bude dít v samotném programu a co k tomu bude potřeba. Je zde probrána modifikovaná metoda uzlových napětí, která se využívá při návrhu. Dále popis, jakým způsobem, se budou jednotlivé prvky algoritmizovat a je zde i vývojový diagram.

V poslední kapitole je poměrně podrobně rozebráno programové řešení, jak by měl program fungovat. Bohužel, program se mi nepodařilo kompletně celý zprovoznit.

LITERATURA

- [1] BIOLEK, D. *Řešíme elektronické obvody aneb kniha o jejich analýze*. 1. vyd. Praha: BEN-Technická literatura, 2004. ISBN 80-7300-125-X.
- [2] BIOLEK, D. Grafy signálových toků pro analýzu obvodů (nejen) v proudovém módu. *Elektrorevue - Internetový časopis* [online]. 2002, č. 31, s. 1 [cit. 2012-11-18]. ISSN 1213- 161X. Dostupné z: <http://www.elektrorevue.cz/clanky/02031/index.html>
- [3] ČAJKA, J., T. DOSTÁL a K. VRBA. Transformace umožňující získat přidružené obvody v proudovém módu. *Elektrorevue - Internetový časopis* [online]. 2000, č. 24, s. 4 [cit. 2012-11-19]. ISSN 1213-1539. Dostupné z: <http://www.elektrorevue.cz/clanky/00023/index.html>
- [4] DOSTÁL, Tomáš. *Elektrické filtry*. 1. vyd. Brno: FEI VUT, 2001. 96 s. ISBN 80-214-0877-4.
- [5] GREGOŘICA, M. Návrh multifunkčního obvodu s proudovými konvejory druhé generace. *Elektrorevue - Internetový časopis* [online]. 2002, č. 56 [cit. 2012-11-18]. Dostupné z: <http://www.elektrorevue.cz/clanky/02056/index.html>.
- [6] HÁJEK, Karel. *Kmitočtové filtry*. 1. vyd. Praha: BEN - technická literatura, 2002, 535 s. ISBN 80-730-0023-7.
- [7] KOTON, J., VRBA, K. Zobecněné metody návrhu kmitočtových filtrů. *Elektrorevue - Internetový časopis* [online]. 2008, č. 26 [cit. 2012-11-18]. Dostupné z: <http://www.elektrorevue.cz/cz/download/zobecnene-metody-navrhu-kmitoctovych-filtru/>.
- [8] KOTON, J., K VRBA, P. USHAKOV a J. MIŠUREC. Návrh elektronicky říditelných kmitočtových filtrů využitím teorie grafů signálových toků. In: *Proceedings of the 31th International Conference Telecommunications and Signal Processing, TSP 2008*. Budapešť: Asszisztencia Szervezo Kft., 2008, s. 3. ISBN 978-963-06-5487-6.
- [9] *Modifikovaná metoda uzlových napětí s refukcí počti rovnic*. 16 s. Dostupné z: http://hippo.feld.cvut.cz/vyuka/aeo/soubory/aeo_mmun.pdf.
- [10] VRBA, K. a J. ČAJKA. *Nové typy proudových konvektorů a jejich aplikace*. Výzkumná zpráva. Brno: 2000, s. 25.

- [11] VRBA, K. *Universální aktivní obvodové prvky a jejich aplikace*. Výzkumná zpráva, 2004.
- [12] ŠPONAR, R. Syntetické dvojpólové prvky s imitancemi vyšších řádů v kmitočtových filtrech s proudovými konvejory. *Elektrorevue - Internetový časopis* [online]. 2004, roč. 3, č. 5, s. 15 [cit. 2012-11-20]. ISSN 1213-1539. Dostupné z: <http://www.elektrorevue.cz/clanky/04013/index.html>
- [13] REICHL, J. Úvod do softwaru Mathematica. *Matematica - forum* [online]. 2012, [cit. 2013-5-20]. Dostupné z: <http://www.mathematica-forum.cz/materialy.htm>
- [14] Wolfram Research, Inc. *Wolfram Reasearch: Mathematica, Technical and Scientific Software* [online]. 2012, [cit. 2013-5-20]. Dostupné z: <http://www.wolfram.com>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

MO-CF vícevýstupový proudový sledovač (Multiple-Output Current Follower)

CCII- proudový konvektor

FNDR Kmitočtově závislý negativní rezistor

\mathbf{I} vektor budících proudů

$\mathbf{p} = \mathbf{j}\omega$ komplexní proměnná

D, E elementární dvojbrany

N_D, N_E řád elementárních dvojbranů

\mathbf{Y} admitanční matice

\mathbf{U} vektor uzlových napětí

x, y, z konstanty

VCCS napětím řízený zdroj proudu

SEZNAM PŘÍLOH

Zdrojové kódy Mathematici a vstupní soubor pro ně(přiloženy na CD)

- HlavniProgram.nb
- navyseniM.nb
- NoveZadani.nb
- zadani.txt

Bakalářská práce v elektronické podobě (přiložena na CD)

- BP.pdf